LISTING OF CLAIMS

1. (currently amended) A program optimization method for translating, into machine code, source code for a program written in a programming language, and for <u>dynamically</u> optimizing said program comprising the steps of:

performing ~~an~~ <u>a dynamic</u> analysis <u>during execution</u> to determine whether the execution speed of said program can be increased by fixing, in a specific state, a parameter for a predetermined command in said program; and

employing results of said analysis for the <u>dynamic</u> generation, in said program, of a path along which said parameter of said predetermined command is fixed in said specific state.

2. (currently amended) The program optimization method according to claim 1, wherein said step of generating a path includes the steps of:

executing said program and obtaining statistical data for the appearance frequency of each available state wherein, according to said results of said analysis, said parameter of said predetermined command may be set; and

employing said obtained statistical data to <u>dynamically</u> generate said path.

3. (currently amended) A program optimization method for translating, into machine code, the source code for a program written in a programming language, and for optimizing said program comprising the steps of:

executing a program to obtain statistical data for an appearance frequency of each available state in which a

JP920000420US1   -2-

parameter of a predetermined command in said program may be set; and

employing said obtained statistical data to dynamically generate a machine language program that includes, as the compiling results, a path along which said parameter of said predetermined command is fixed in a specific state.

4. (original) The program optimization method according to claim 3, further comprising a step of:

generating a machine language program that does not include, as a compiling result, a path along which said parameter of said predetermined command is fixed in a specific state.

5. (currently amended) A program optimization method for translating, into machine code, the source code for a program written in an object-oriented programming language, and for optimizing said program comprising the steps of:

detecting one command dynamically during execution, of the commands in said program, for which a method call destination can be identified, and for which the processing speed can be increased by identifying said method call destination; and

dynamically generating a path wherefor said method call destination for said detected command is limited in order to increase the processing speed of said command.

6. (currently amended) A program optimization method for translating into machine code, the source code for a program written in a programming language, and for optimizing said program comprising the steps of:

JP920000420US1   -3-

detecting <u>dynamically during program execution</u> one command, of the commands in said program, for which a variable can be limited to a predetermined constant value, and for which the processing speed can be increased by limiting said variable to said constant value; and

generating a path along which said constant value said variable of said detected command is fixed.

7. (currently amended) A <u>dynamic</u> compiler for translating into machine code the source code for a program written in a programming language, and for optimizing the resultant program comprising:

an impact analysis unit for performing an analysis to <u>dynamically</u> determine <u>during execution</u> how much the execution speed of said program can be increased by fixing, in a specific state, a parameter of a predetermined command in said program; and

a specialization unit for employing the analysis results obtained by said impact analysis unit to generate, in said program, a specialized path along which said parameter of said predetermined command is fixed in said specific state.

8. (original) The compiler according to claim 7, further comprising:

a data specialization selector for, when said program is executed, obtaining statistical data for the appearance frequency of each state obtained by said impact analysis unit, and for determining the state in which said parameter of said predetermined command is to be set, wherein said specialization unit generates a specialized path along which

JP920000420US1   ~4~

said parameter of said predetermined command is fixed in a state determined by said data specialization selector.

9. (original) The compiler according to claim 8, wherein, in accordance with the state of said program at execution, said specialization unit generates, in said program, a branching process for selectively performing a specialized path and an unspecialized path; and wherein, while taking into account a delay due to the insertion of said branching process, said data specialization selector determines a state in which said parameter of said predetermined command is fixed.

10. (currently amended) A computer comprising:
an input device for receiving source code for a program;

a <u>dynamic</u> compiler for translating said source code to compile said program and for converting said compiled program into machine language code; and

a processor for executing said machine language code,
wherein said <u>dynamic</u> compiler includes

means for performing <u>a dynamic</u> ~~an~~ analysis to determine <u>during execution</u> whether the execution speed of said program can be improved by fixing in a specific state a parameter of a predetermined command in said program, and

means for generating in said program, based on the analysis results, a path along which said parameter of said predetermined command is fixed in said specific state and for compiling said program, and

wherein said compiler outputs, as the compiled results, said machine language code that includes said path along which the state of said parameter is fixed.

JP920000420US1   -5-

11. (currently amended) A computer comprising:

    an input device, for receiving source code for a
program;

    a <u>dynamic</u> compiler, for translating said source code to
compile said program and for converting said compiled
program into machine language code; and

    a processor, for executing said machine language code,
    wherein said <u>dynamic</u> compiler includes

        means for obtaining statistical data for the
appearance frequency of each available state wherein a
parameter for a predetermined command in said program may be
set when said program is executed, and for employing said
statistical data to determine a state in which said
parameter of said predetermined command is to be fixed, and

        means for generating a specialized path along
which said parameter of said predetermined command is fixed
in said determined state, and for compiling said program,
and

    wherein said compiler outputs, as the compiled results,
said program as said machine language code that includes
said specialized path.

12. (original) The computer according to claim 11,
comprising:

    said compiler further includes

        means for compiling said program without
generating a specialized path,

JP920000420US1   —6—

wherein, when said state of said parameter to be fixed can not be determined, said means for determining the state of said parameter of said predetermined command outputs, as compiled results, said program in said machine language code, which is generated by said means for compiling said program without generating said specialized path, that does not include said specialized path.

13. (currently amended) A support program, for controlling a computer to support generation of a program, which permits said computer to perform:

a function for performing a dynamic an analysis to determine during execution whether the execution speed of said program can be increased by fixing a parameter of a predetermined command of said computer program in a specific state; and

a function for generating in said program, based on the analysis results, a path along which said parameter of said predetermined command is fixed in said specific state.

14. (original) A support program, for controlling a computer to support generation of a program, which permits said computer to perform:

a function for executing said program and obtaining statistical data for the appearance frequency of each available state wherein said parameter of said predetermined command of said program may be set; and

a function for generating in said program, based on said statistical data, a path along which said parameter of said predetermined command is fixed in said specific state.

JP920000420US1   -7-

15. (currently amended) A storage medium on which input means of a computer stores a computer-readable support program, for controlling said computer to support generation of a program, that permits said computer to perform:

a function for performing a dynamic ~~an~~ analysis to determine during execution whether the execution speed of said program can be increased by fixing a parameter of a predetermined command of said computer program in a specific state; and

a function for generating in said program, based on the analysis results, a path along which said parameter of said predetermined command is fixed in said specific state.

16. (original) A storage medium on which input means of a computer stores a computer-readable support program, for controlling said computer to support generation of a program, that permits said computer to perform:

a function for executing said program and obtaining statistical data for the appearance frequency of each available state wherein said parameter of said predetermined command of said program may be set; and

a function for generating in said program, based on said statistical data, a path along which said parameter of said predetermined command is fixed in said specific state.

JP920000420US1   -8-